

MENU

SEARCH

INDEX

DETAIL

BACK

2/2



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 10282881

(43)Date of publication of application: 23.10.1998

(51)Int.CL

G09C 1/00

G09C 1/00

G09C 1/00

H04L 9/08

H04L 9/30

(21)Application number: 09086302

(71)Applicant:

NIPPON TELEGR & TELEPH
CORP <NTT>

(22)Date of filing: 04.04.1997

(72)Inventor:

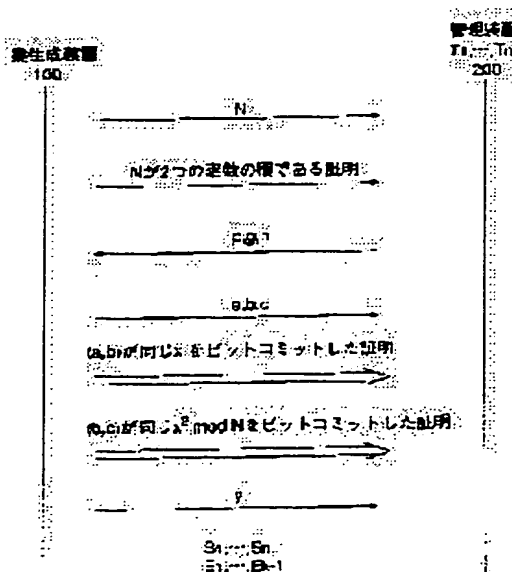
OKAMOTO TATSUAKI

(54) SECRET KEY DECENTRALIZATION MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To restore a secret key by gathering a certain number of pieces of decentralized registration information for a prime factor factorization system open key cipher.

SOLUTION: A key generator K100 sends an open key N and information providing that N is the product of two prime numbers to an administrator T_i ($i=1\dots n$) 200, and T sends a prime number (p) and elements (g) and (h) whose order is N to K; and K generates random numbers (x) and r_1 , r_2 , and r_3 , calculates $a=BC_e(x, r_1)$, $b=BC(x, r_2)$, and $c=BC_b(x_2 \bmod N, r_3)$ with a bit commitment function BC and sends them to T_i , and K prove that (a) and (b) made a bit commitment to the same (x) and (b) and (c) made a bit commitment to the same $x_2 \bmod N$ by exchanging



with $T+i$, sends (y) satisfying $y^2 \equiv (x) \pmod{N}$ ($y/N \neq -1$) to T_i , disperses $8x$ to $s_1 \dots s_n$ and sends s_i to corresponding T_i , thereby sending pieces of information $E_1 \dots E_{k-1}$ providing that s_i is a correct value.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

[MENU](#)[SEARCH](#)[INDEX](#)[DETAIL](#)[BACK](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-282881

(43) 公開日 平成10年(1998)10月23日

(51) IntCl. ⁶	識別記号	F I	
G 0 9 C 1/00	6 3 0	G 0 9 C 1/00	6 3 0 D
	6 2 0		6 2 0 B
	6 5 0		6 5 0 Z
H 0 4 L 9/08		H 0 4 L 9/00	6 0 1 D
9/30			6 6 3 B
審査請求 未請求 請求項の数2 O L (全 7 頁)			

(21) 出願番号 特願平9-86302

(22) 出願日 平成9年(1997)4月4日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 岡本 龍明

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

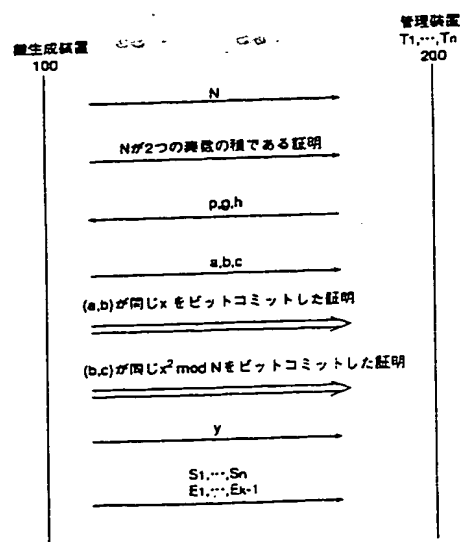
(74) 代理人 弁理士 草野 卓

(54) 【発明の名称】 秘密鍵分散管理方法

(57) 【要約】

【課題】 素因数分解系公開鍵暗号に対し、分散登録情報がある一定数集めることで秘密鍵を復元可能とする。

【解決手段】 鍵生成者Kは公開鍵Nと、Nが2つの素数の積であることを証明する情報を管理者 T_i ($i = 1, \dots, n$)へ送り、 T_i は素数 p と位数がNの元 g , h をKへ送り、Kは乱数 x と r_1, r_2, r_3 を生成し、ビットコミットメント関数BCで $a = BC_g(x, r_1)$, $b = BC_a(x, r_2)$, $c = BC_b(x^2 \bmod N, r_3)$ を演算して T_i へ送り、Kは T_i とのやりとりで a と b が同じ x をビットコミットしたものの、また b と c が同じ $x^2 \bmod N$ をビットコミットしたものであることを証明し、 $y^2 \equiv x^4 \pmod{N}$ ($y/N = -1$)を満す y を T_i へ送り、 x を s_1, \dots, s_n に分散して s_i を対応 T_i へ送りまた s_i が正しい値であることを証明する情報 E_1, \dots, E_{k-1} を送る。



【特許請求の範囲】

【請求項1】 鍵生成者の鍵生成装置と、管理者 T_i

($i=1, \dots, n$)の管理装置とにより、素因数分解問題に基づく公開鍵暗号方式の秘密鍵を複数に分散して上記管理装置に登録し後に復元する秘密鍵分散管理方法であって、

鍵生成者は鍵生成装置により秘密鍵 (P, Q) と公開鍵 $N=PQ$ を生成し、 N を管理者 T_i ($i=1, \dots, n$)の管理装置に送り管理装置は単独又は複数共同で素数 p ならびに位数が N の元 g を定めて鍵生成装置へ送り、鍵生成装置は $y, a = F_g(x) \bmod p$ を管理装置に送るとともに、 N が2つの素数の積であることならびに x と y が一定の関係を満足することを x, P, Q の値を秘密にして管理装置との間の情報交換により、管理者に証明し、

鍵生成装置は x をShamirの多項式補間法を用いて n 個の値 s_1, \dots, s_n に分散し、 s_i を管理者 T_i の管理装置に送ると同時に、 s_i が正しく分散された値であることを証明する情報を送り、

管理装置はそれらの正当性を検証し、正しければそれらを受けとり保管し、秘密鍵 (P, Q) を復元する必要があるときには n 人の管理者の中で k 人が協力して、各管理者の分散情報をその管理装置より鍵生成装置に供給し、鍵生成装置はこれら供給された分散情報より x の値を復元し、それと y の値より N の素因数 P, Q を計算することを特徴とする秘密鍵分散管理方法。

【請求項2】 鍵生成者の鍵生成装置と、管理者 T_i

($i=1, \dots, n$)の管理装置によりRSA公開鍵暗号方式の秘密鍵を複数に分散して上記管理装置に登録し、後に復元する秘密鍵分散管理方法であって、

鍵生成者は鍵生成装置によりRSA公開鍵暗号方式の秘密鍵 d と公開鍵 (N, e) を生成し、 (N, e) を管理者 T_i ($i=1, \dots, n$)の管理装置に送り、管理者は単独又は複数共同で管理装置により乱数 X を生成すると共に $G = X \bmod N$ を計算し、また素数 p ならびにパラメータ g を生成し、 X, p, g を鍵生成装置に送り、

鍵生成装置は $a = F_g(d) \bmod p$ を管理装置に送るとともに、鍵生成者は a と $X = F'_g(d) \bmod N$ がそれぞれ同じ d を暗号化していることを d の値を秘密にしたまま管理者鍵生成装置と管理装置との情報交換により証明し、

鍵生成装置は d をShamirの多項式補間法を用いて n 個の値 s_1, \dots, s_n に分散し、 s_i を管理者 T_i の管理装置に送ると同時に、 s_i が正しく分散された値であることを証明する情報を送り、

管理装置はそれらの正当性を検証し、正しければそれらを受けとり保管し、

秘密鍵 d を復元する必要があるときには n 人の管理者の中で k 人が協力して、各管理者の分散情報をその管理装

置より鍵生成装置へ供給し、鍵生成装置は供給された分散情報より d の値を復元することを特徴とする秘密鍵分散管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、公開鍵暗号の秘密鍵を分散して登録しておき、紛失時やその他の理由により秘密鍵を復元する必要があるときに、ある一定以上の数の分散登録情報を集めることにより秘密鍵を復元する秘密鍵分散管理方法に関するものである。

【0002】

【従来の技術】従来、単に情報を分散管理し復元する方式はShamirの多項式補間法(“Howto Share a Secret”, Comm. Assoc. Comput. Mach., vol.22, no.11, pp. 612-613 (Nov.1979))があるが、分散情報が正しい情報であることを証明する手段がないため、そのままでは秘密鍵を分散して登録する手段には使えない。分散情報が正しい情報であることを証明する方式としては、PedersenによるVSSと呼ばれる方法がある(“Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”, Proc.of Crypto'91, LNCS 576, Springer-Verlag, pp.129-140(1992))。しかし、この方法ではElGamal法などの離散対数問題系の公開鍵暗号には適用できるが、RSA法などの素因数分解系の公開鍵暗号には適用できない。

【0003】一方、Micaliにより(“Fair Public-Key Cryptosystems”, Proc.of Crypto'92, LNCS, Springer-Verlag, pp.113-138(1993))RSA法などに適用できる秘密鍵の分散管理/復元方式が提案されているが、管理者が全員協力しないと秘密鍵を復元できない(つまり、一人でも協力しないと秘密鍵を復元できなくなる)。

【0004】

【発明が解決しようとする課題】この発明の目的は、素因数分解系の公開鍵暗号に対して秘密鍵を分散して登録しておき、ある一定以上の数の分散登録情報を集めることにより秘密鍵を復元する方法を実現することにある。

【0005】

【課題を解決するための手段】この発明では、秘密鍵もしくは秘密鍵と等価な秘密情報を例えばビットコミットメントとよばれる手法で暗号化し、そこで暗号化された秘密情報が正しい情報であることを証明する効率的な方法を開発した。一旦正しい秘密情報を暗号化したビットコミットメントが得られると、前述したPedersenの手法を用いて、分散情報が正しい情報であることを証明することが可能である。

【0006】請求項1では、 y を公開しておき、公開鍵 N を素因数分解するための秘密情報として $x^4 \equiv y^2 \pmod{N}$ であるような x を用いる。請求項2では、RSA暗号の秘密鍵 d をビットコミットメントの対象とす

る。

[0007]

【発明の実施の形態】以下では、この発明の請求項1の一実施例について説明する。図1はこの発明の全体構成を示す。鍵生成者の装置（鍵生成装置）100は、管理者 T_i の装置（管理装置）200 i ($i=1, 2, \dots, n$)とそれぞれ、通信路300 i を介して結合されているとする。図2にこの発明の通信シーケンス例を示し、以下、それぞれ図3に鍵生成装置100の機能構成例を、図4に管理装置200 i の機能構成例を示す。

[0008] 以下、分散情報の登録ならびに秘密鍵を回復する手順を示す。

1. 鍵生成者は鍵生成装置100内の鍵生成器101を用いて、秘密鍵 (P, Q) と公開鍵 $N=PQ$ を生成し、 N を管理者 T_i の管理装置200 i ($i=1, \dots, n$)に送る。ここでヤコビ記号の値が $(-1/N)=1$ とする。これは N を素因数 P, Q に分解できる条件である。

[0009] 2. 鍵生成装置100の剰余演算器102を用いて N が2つの素数の積であることを証明する情報を生成し、それを管理装置200 i に送る。このような情報の作成方法は、Micaliの方法（“Fair Public-Key Cryptosystems”, Proc.of Crypto'92, LNCS, Springer-Verlag, pp.113-138(1993)）の中で示されている。

3. 管理装置200 i は単独又は共同でパラメータ生成器201を用いて素数 p ならびに位数が N の元 g, h を定め、それを鍵生成装置100に送る。ここで、ビットコミットメント関数 BC を以下のように定義する。

[0010] $BC_g(x, r) = g^x h^r \bmod p$ 4. 鍵生成装置100は乱数生成器103を用いて乱数 x ならびに適当な値 r_1, r_2, r_3 を定め、剰余演算器104を用いて次の a, b, c を計算し、管理装置200 i に送る。

$$a = BC_g(x, r_1)$$

$$b = BC_a(x, r_2) = BC_g(x^2 \bmod N, r_1 x + r_2 \bmod N)$$

$$c = BC_b(x^2 \bmod N, r_3) = BC_g(x^4 \bmod N, r_1 x^3 + r_2 x^2 + r_3 \bmod N)$$

5. 次に、鍵生成装置100は $a = BC_g(x, r_1)$ と $b = BC_a(x, r_2)$ が同じ x をビットコミットしたものであることを以下の手順で証明する。

[0011] (a) 鍵生成装置100は乱数生成器103を用いて v, w, w' を生成し、さらに剰余演算器105を用いて

$$u = BC_g(v, w), u' = BC_a(v, w')$$

を生成し、 (u, u') を管理装置200 i に送る。

(b) 管理装置200 i は乱数生成器202を用いて R を定め、それを鍵生成装置100に送る。

[0012] (c) 鍵生成装置100は剰余演算器106を用いて

$$z = v + xR \bmod N, t = w + r_1 R \bmod N, t' =$$

$$w' + r_2 R \bmod N$$

を計算し、 (z, t, t') を管理装置200 i に送る。

(d) 管理装置200 i は以下の式が成立するかどうかを剰余演算器203ならびに比較器204を用いて検証する。

$$[0013] BC_g(z, t) = u a^R \bmod p, BC_a(z, t') = u' b^R \bmod p$$

$$6. \text{ 鍵生成装置100は同様に}$$

$$10 \quad b = BC_g(x^2 \bmod N, r_1 x + r_2 \bmod N) \\ c = BC_b(x^2 \bmod N, r_3)$$

が同じ $x^2 \bmod N$ をビットコミットしたものであることを上と同じ手順で証明する。

[0014] 7. 鍵生成装置100は剰余演算器107を用いて $y^2 \equiv x^4 \pmod{N}$ およびヤコビ記号の値が $(y/N) = -1$ を満足する y を計算し管理装置200 i に送る。ここで y は x と $y^2 = x^4 \pmod{N}$ の関係にあることが前記第5ステップと第6ステップの両証明により証明されており、かつ $(y/N) = -1, x, y$ が与えられると N は素因数 P, Q に分解できる条件である。

[0015] 8. 鍵生成装置100は x をShamirの多項式補間法により剰余演算器108を用いて n 個の値 s_1, \dots, s_n に分散し、 s_i を管理者 T_i の管理装置200 i に送る。さらに、 $a = BC_g(x, r_1)$ の値を利用して、Pedersenの方法により剰余演算器108を用いて s_i が正しく分散された値であることを証明する情報 E_1, \dots, E_{k-1} を送る。 $k-1$ は前記多項式補間法に用いる多項式の最低の次数であり、 $k \leq n$ である。

30 [0016] 9. 管理装置200 i は剰余演算器205および比較器206を用いてそれらの正当性を検証し、正しければそれらを受けとり保管する。

10. 秘密鍵 (P, Q) を復元する必要があるときには n 人の管理者の中で k 人が協力して、各管理者の分散情報 s_i を持ち寄りそれらよりShamirの方法により剰余演算器109を用いて x の値を復元する。さらに、GCD演算器110を用いてGCD演算器110を用いて、 $x^2 - y$ と N との最大公約数を求めると、 P 又は Q が求まり、これより、 P, Q の両者を求めることができる。つまり、いま $m = x^2 \pmod{N}$ 、ヤコビ記号の値が $(m/N) = 1$ とすると、 $m^2 = y^2 = x^4 \pmod{N}$ であり、かつ $(1/N) = -1, (y/N) = -1$ の条件がある場合は、 $(z - y)$ 、つまり $(x^2 - y)$ は N の素因数で割り切れることが数学的に知られている。よって $x^2 - y$ と N との最大公約数を求めれば N の1つの素因数が得られることになる。

[0017] 次に、この発明の請求項2の一実施例について説明する。図5にこの発明の通信シーケンス例を示し、以下、それぞれ、図6に鍵生成装置100の構成例を、図7に管理装置200 i の構成例を示す。以下、分

散情報の登録ならびに秘密鍵を回復する手順を示す。

1. 鍵生成装置100は鍵生成器101を用いて、RSA暗号の秘密鍵 d と公開鍵 (N, e) を生成し、 (N, e) を管理者 T_i ($i=1, \dots, n$)の管理装置200*i*に送る。

【0018】2. 管理装置200*i*は単独又は共同でパラメータ生成器201を用いて素数 (p, q) ならびに位数が q の元 g, h を定め、それを鍵生成装置100に送る。なお、 $q \mid p-1$ (q は $p-1$ の約数)とする。

ここで、ビットコミットメント関数 BC を以下のように定義する。 $BC_g(x, r) = g^x h^r \bmod p$ 10
さらに管理装置200*i*は乱数生成器202を用いて X を生成し、剰余演算器203を用いて $G = X^e \bmod N$ を計算する。

【0019】管理装置200*i*は X ならびに (p, q, g, h) を鍵生成者に送る。ここで、 $BC_g(s, r) = g^s h^r \bmod p$, $BC_g(s) = G^s \bmod N$ とする。

3. 鍵生成装置100は乱数生成器103を用いて適当な値 r_1 を定め、剰余演算器102を用いて
 $a = BC_g(d, r_1)$
を計算するとともに
 $b = BC_g(d) = X$
とする。

【0020】4. 次に、鍵生成装置100は $a = BC_g(d, r_1)$ と $b = BC_g(d) = X$ が同じ d をビットコミットしたものであることを以下の手順で証明する。

(a) 鍵生成装置100は乱数生成器103を用いて v, w を生成し、さらに剰余演算器104を用いて
 $u = BC_g(v, w)$, $u' = BC_g(v)$
を生成し、 (u, u') を管理装置200*i*に送る。な 30
お u' の計算に用いる G は X, e, N により計算する。

【0021】(b) 管理装置200*i*は乱数生成器202を用いて R を定め、それを鍵生成装置100に送る。

(c) 鍵生成装置100は、剰余演算器105を用いて
 $z = v + dR$, $t = w + r_1 R \bmod q$
を計算し、 (z, t) を管理装置200*i*に送る。

【0022】(d) 管理装置200*i*は以下の式が成立するかどうかを剰余演算器204ならびに比較器205を用いて検証する。

$$BC_g(z, t) = u a^R \bmod p, \quad BC_g(z) = u' b^R \bmod N \quad 40$$

さらに比較器205を用いて z が v, R, d のサイズ(ビット数)で決まるビット数以下であることを確認する。

【0023】5. 鍵生成装置100は d をShamirの多項式補間法により剰余演算器106を用いて n 個の値 s_1, \dots, s_n に分散し、 s_i を管理者 T_i の管理装置200*i*に送る。さらに、 $a = BC_g(d, r_1)$ の値を利用して、Pedersenの方法により剰余演算器106を用いて s_i が正しく分散された値であることを証明する情報 E_1, \dots, E_{k-1} を送る。

【0024】6. 管理装置200*i*は剰余演算器206および比較器207を用いてそれらの正当性を検証し、正しければそれらを受けとり保管する。

7. 秘密鍵 d を復元する必要があるときには n 人の管理者の中で k 人が協力して、各管理者の分散情報を持ち寄りそれらよりShamirの方法により剰余演算器107を用いて d の値を復元する。

【0025】上述では請求項1の発明で $a = F_g(x) \bmod p$ として $g^x h^r \bmod p$ を用いたが $g^x \bmod p$ でもよい。同様に請求項2の発明でも $a = F_g(d) \bmod p$ として $g_d \bmod p$ でもよい。

【0026】

20 【発明の効果】請求項1の発明では、素因数分解系の公開鍵暗号であるRSA法やRabin法に対して秘密鍵

(P, Q) に関連する秘密情報 x を分散して登録しておく、しかも各管理者は自分に送られてきた分散情報が正しいことを確認できる。従って、一定人数(k 人)の管理者が協力すれば必ず秘密鍵 (P, Q) を復元できる。また、同様に請求項2の発明では、RSA法に秘密鍵 d を分散管理し一定人数(k 人)の管理者が協力すれば必ず秘密鍵 d を復元できる。

【図面の簡単な説明】

30 【図1】この発明が適用されるシステムの構成を示すブロック図。

【図2】請求項1の発明の実施例における鍵生成装置と管理装置間で行う処理シーケンスを示す図。

【図3】図2の実施例における鍵生成装置100の機能構成を示すブロック図。

【図4】図2の実施例における管理装置200*i*の機能構成を示すブロック図。

【図5】請求項2の発明の実施例における鍵生成装置と管理装置間で行う処理シーケンスを示す図。

【図6】図5の実施例における鍵生成装置100の機能構成を示すブロック図。

【図7】図5の実施例における管理装置200*i*の機能構成を示すブロック図。

【図1】

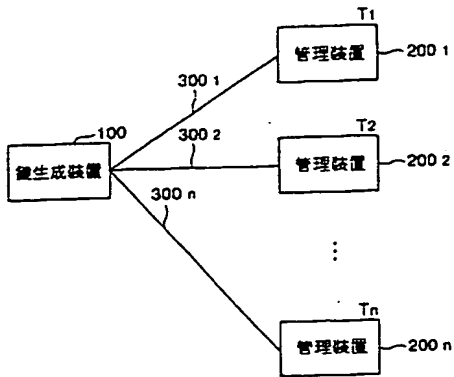


図 1

【図2】

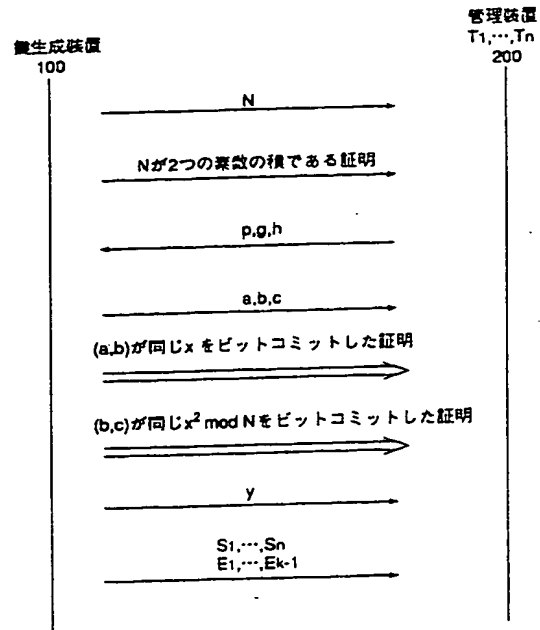


図 2

【図4】

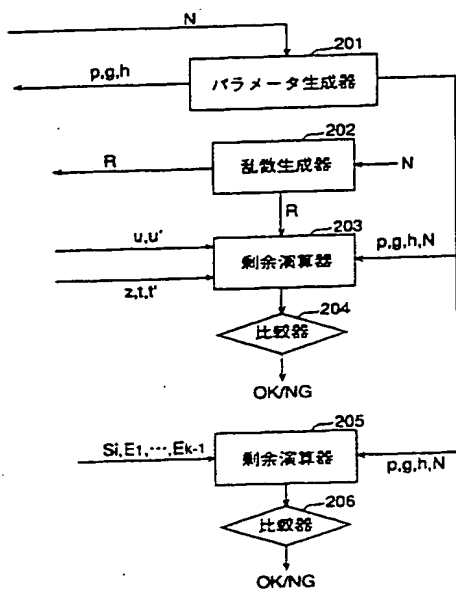


図 4

【図3】

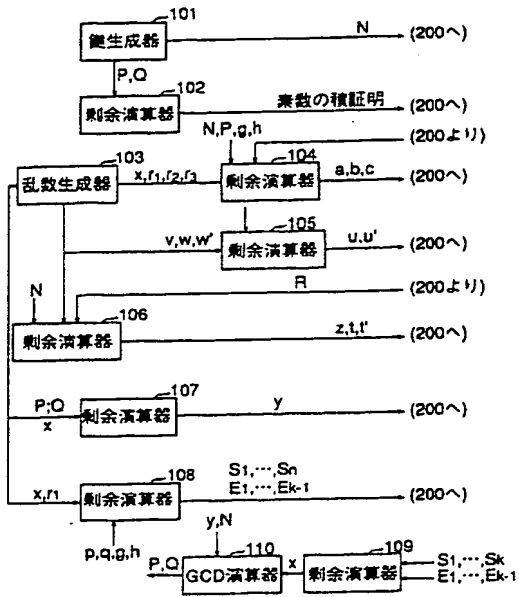


図 3

【図5】

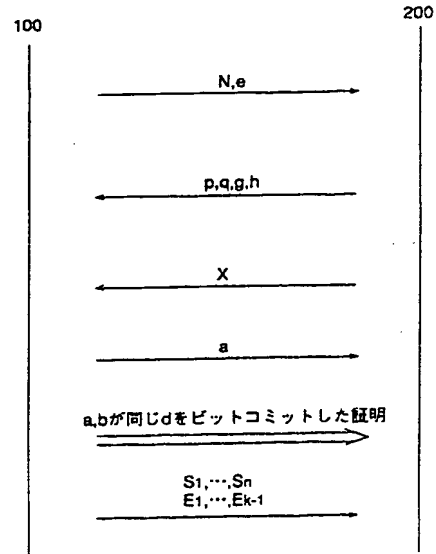


図 5

【図6】

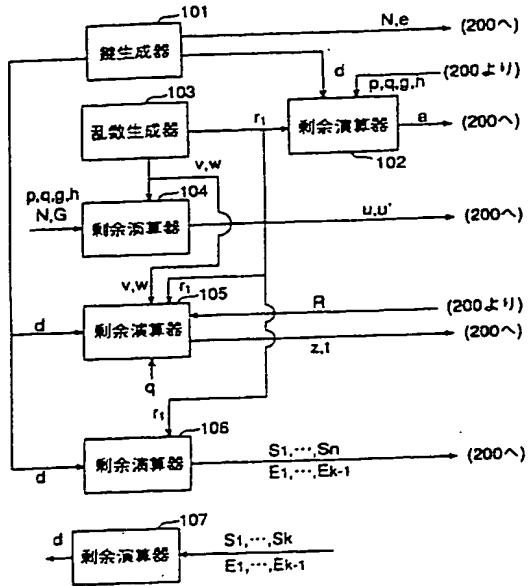


図 6

【図7】

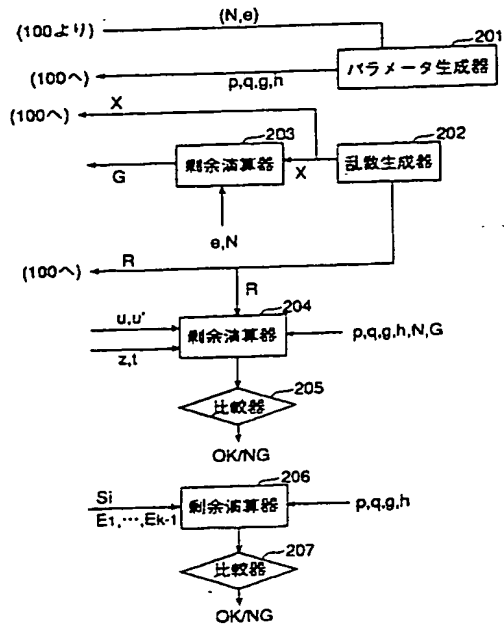


図 7

THIS PAGE BLANK (USPTO)